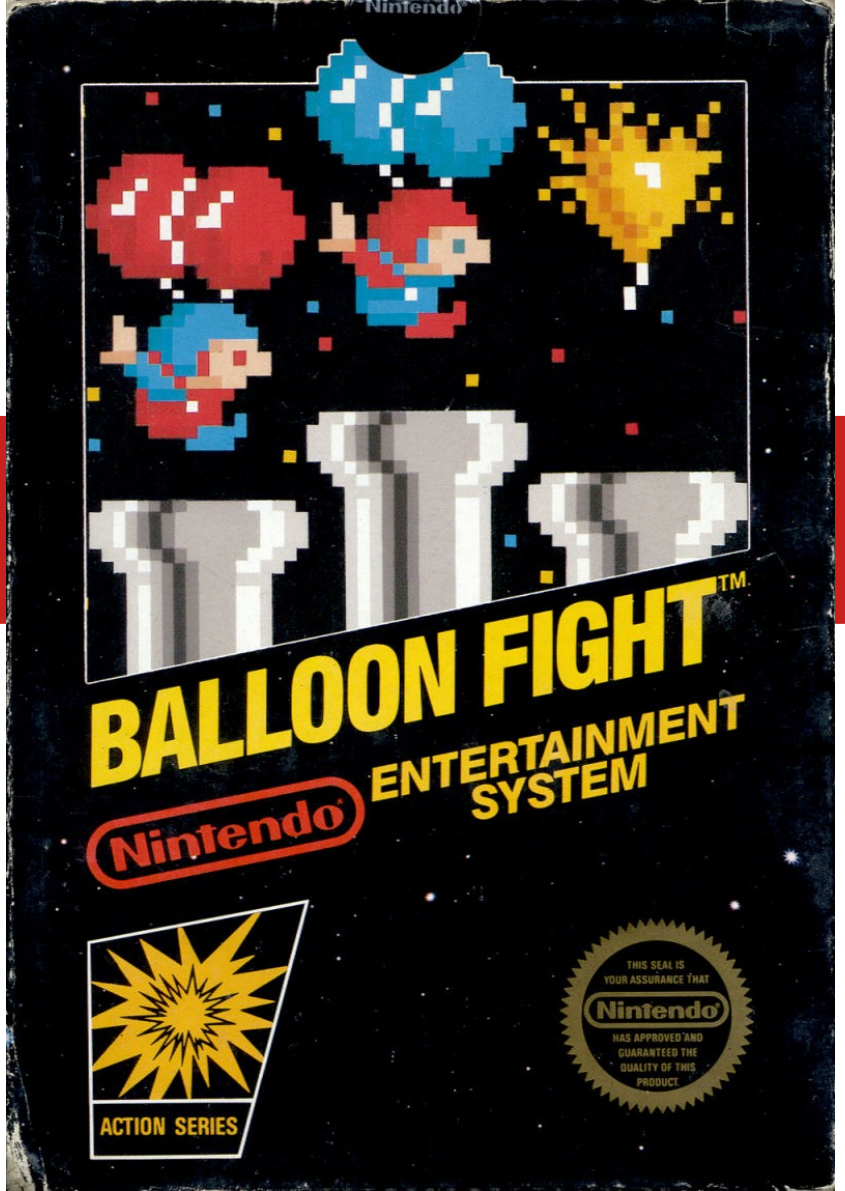
BALLOON FIGHT

Rita Malaquias 110075

Tópicos de Programação para Jogos 2021/2022
Com a orientação de Diogo Gomes



CONTEXTO

- Videojogo originalmente lançado em 1984 para a NES (Nintendo Entertainment System) desenvolvido pela Nintendo;
- Programador do jogo foi Satoru Iwata, ex-presidente da Nintendo;
- A escolha deste jogo teve grande influência por ter sido o Iwata a ser o programador.

CONTEXTO


I-002000
TOP-010000


I-054850
TOP-054850


I-000580
TOP-025000
RANK-47

PADRÕES UTILIZADOS

- State Machine
- Game Loop
- Command
- Singleton
- Bytecode

GAME LOOP E STATE MACHINE

```
#game loop
run = True
while run:
    clock.tick(const.FPS) #runs the while loop at the frames defined

    #pause menu appears
    key_pressed = key.get_pressed()
    if key_pressed[K_p]:
        if current_game_state == MainState.pause:
            current_game_state = MainState.running
        else:
            current_game_state = MainState.pause

    #counts amount of enemies remaining
    remaining_enemies = 0
    for enemy in level.enemy_list:
        if enemy.state != EnemyState.defeated:
            remaining_enemies += 1

    #checks if game over screen should appear
    if remaining_enemies == 0 or player.state == PlayerState.dead:
        current_game_state = MainState.game_over
    #checks click on start button in title screen, which starts the game
    if title_screen.start_button.on_click():
        current_game_state = MainState.running
    #checks click on exit button in title screen
    if title_screen.exit_button.on_click():
        current_game_state = MainState.exit
    #checks click on quit button in game over screen
    if game_ui.quit_button.on_click():
        current_game_state = MainState.exit
```

COMMAND

```
You, 6 hours ago | 1 author (You)
class Command:
    def execute():
        raise NotImplemented

You, 6 hours ago | 1 author (You)
class Jump(Command):
    def execute(player):
        player.jump()

You, 6 hours ago | 1 author (You)
class Left(Command):
    def execute(player):
        player.left()

You, 6 hours ago | 1 author (You)
class Right(Command):
    def execute(player):
        player.right()
```

```
You, 6 hours ago | 1 author (You)
class InputHandler:
    def handleInput(self):
        command_list = [] #in order to allo
        keys_pressed = key.get_pressed()
        if keys_pressed[K_LEFT]:
            command_list.append(Left)
        if keys_pressed[K_RIGHT]:
            command_list.append(Right)
        if keys_pressed[K_SPACE]:
            command_list.append(Jump)
        return command_list
```

```
#command pattern
input_handler = InputHandler() #needs to be
command_list = input_handler.handleInput()
for command in command_list:
    command.execute(player)
```

SINGLETON

```
You, 6 hours ago | 1 author (You)
class Player(Character):
    player_instance = None
    #singleton implementation
    @staticmethod
    def get_instance(position = [0.0, 0.0], momentum = [0.0, 0.0], score = 0):
        if Player.player_instance is None:
            Player(position, momentum, score)
        return Player.player_instance

    def __init__(self, position = [0.0, 0.0], momentum = [0.0, 0.0], score = 0):
        self.score = score
        self.state = PlayerState.normal
        Player.player_instance = self
        super().__init__('playerSprite1.png', PLAYER_WIDTH, PLAYER_HEIGHT, positio
```

BYTECODE

```
"background": [
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 4, 0, 0, 6, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 4, 0, 0, 6, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1],
    [1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1],
    [1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7],
    [7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7]
```

```python
f = open(os.path.join('levelmaps', 'level1.json'))
level = json.load(f)
#print(level_test["background"])

#background
grid_background = level["background"]
#foreground
grid_platform = level["foreground"]
```

DIFICULDADES

- A falta de experiência em Python, o que fez com que o desenvolvimento do projeto demora-se mais a ser concluído
- Colisões com inimigos
- Físicas e gravidade

O QUE FICOU POR FAZER

- Adicionar in-game HUD: score
- Adicionar sprites diferentes consoante a direcção dos characters e o seu acontecimento: por exemplo mudar o sprite do enemy quando ele está no modo defeated
- Adicionar pelo menos mais um nível
- Retry menu
- Som

MELHORIAS

- Melhorar colisões
- Recortar os sprites directamente do atlas, em vez de guardar cada sprite numa imagem
- Melhorar organização geral do código
- Arranjar outra solução para a geração de níveis
- Pause Menu

FUTURAMENTE

- Adicionar o resto dos modos de jogo existentes no original
- Adicionar 2 players locais
- Implementação do enemy da água, em que se o player ou o enemy se aproximarem da água, o peixe sobe e come-os (meter imagem do jogo original)
- Adicionar nuvens que atiram raios

DIAGRAMA DE CLASSES

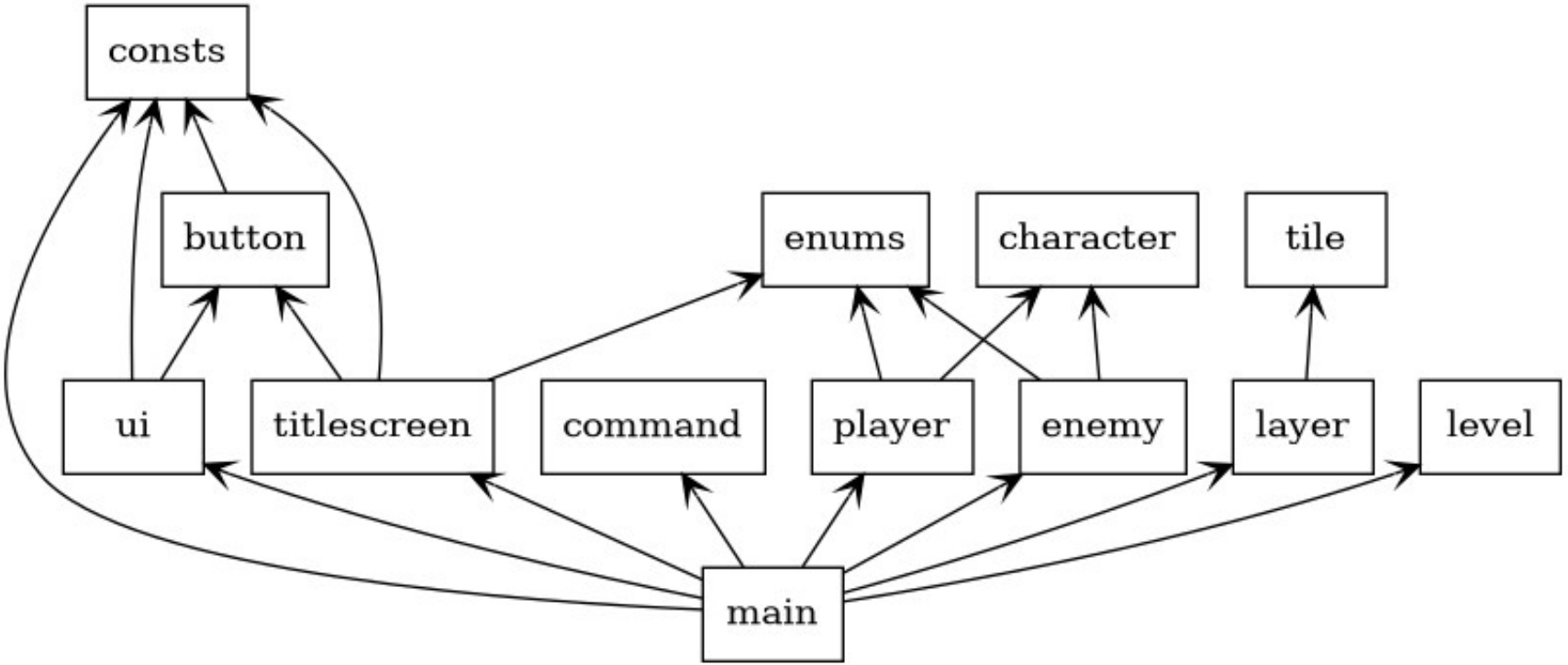
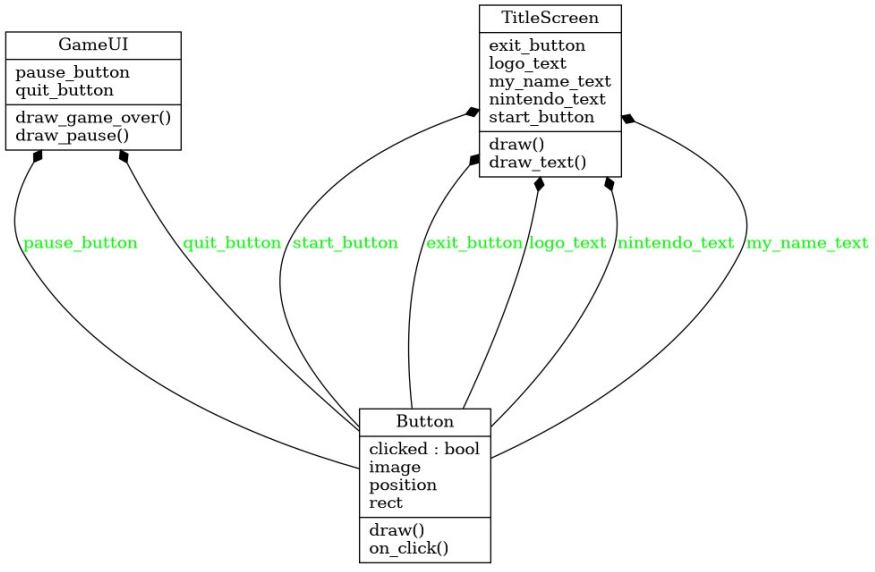


DIAGRAMA DE CLASSES



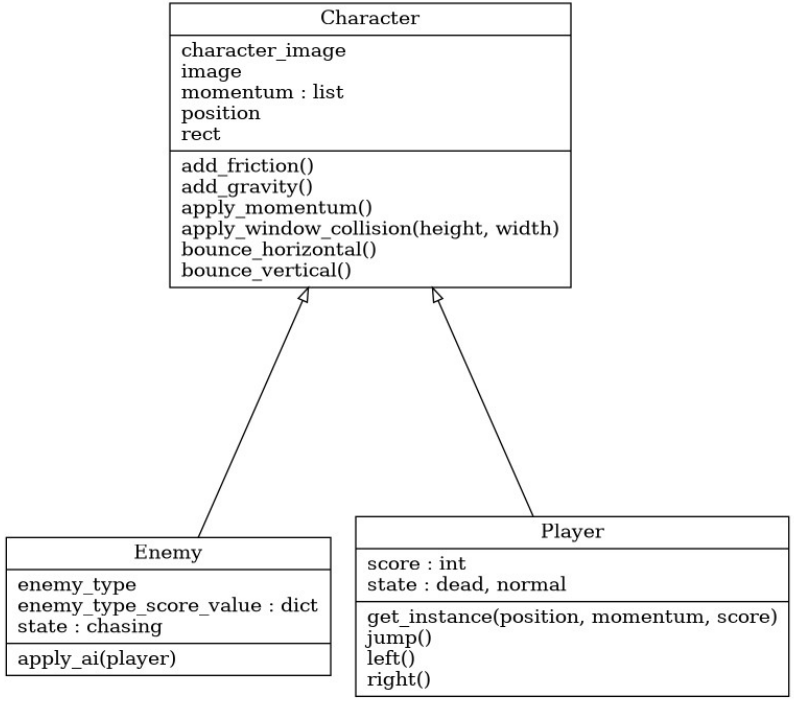
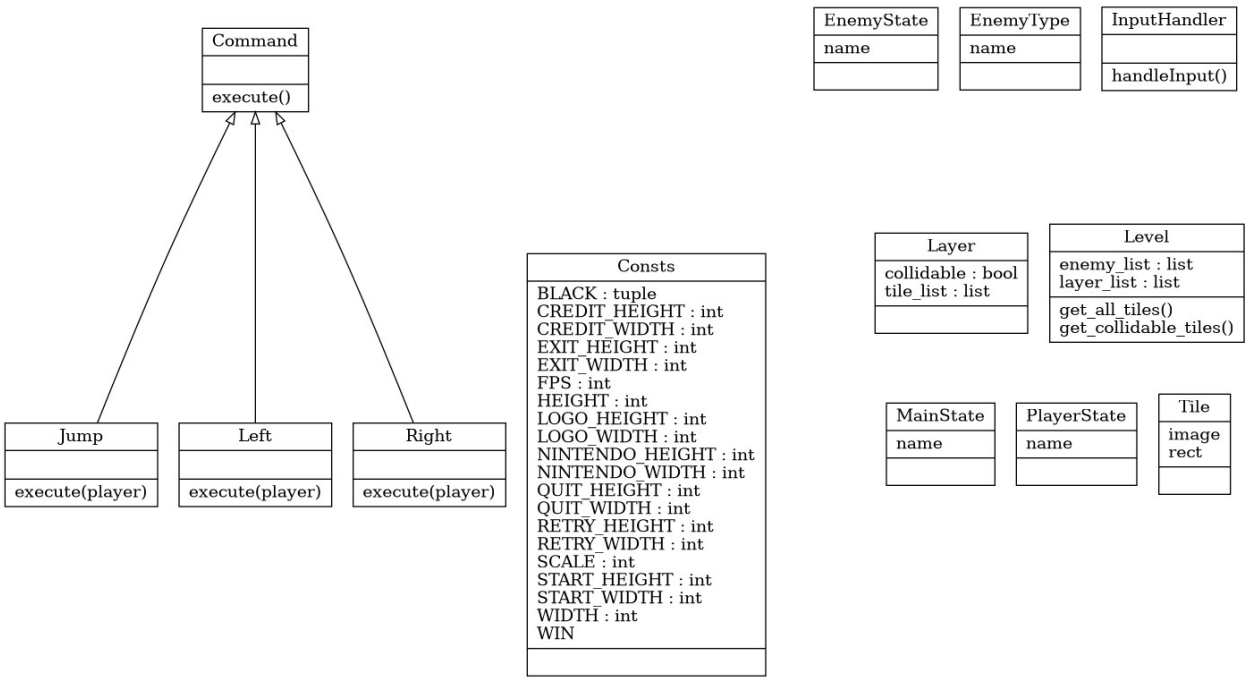


DIAGRAMA DE CLASSES



DEMO

https://github.com/detiuaveiro/projecto-mini-jogo-ritamalaquias